Challenge: Resolving data center power bill disputes: the energy-performance trade-offs of consolidation

Angelos Chatzipapas[+], Dimosthenis Pediaditakis[*], Charalampos Rotsos[*], Vincenzo Mancuso[+], Jon Crowcroft[*], Andrew W. Moore[*]

[*]Computer Laboratory, University of Cambridge, UK {firstname.lastname}@cl.cam.ac.uk

[+]IMDEA Networks Institute, and Universidad Carlos III de Madrid, Spain {firstname.lastname}@imdea.org

ABSTRACT

In this paper we challenge the common evaluation practices used for Virtual Machine (VM) consolidation, such as simulation and small testbeds, which fail to capture the fundamental trade-off between energy consumption and performance. We identify a number of over-simplifying assumptions which are typically made about the energy consumption and performance characteristics of modern networked systems. In response, we describe how more accurate models for data-center systems can be designed and used in order to create an evaluation framework that allows the more reliable exploration of the energy-performance trade-off for VM consolidation strategies.

Categories and Subject Descriptors

C.4 [**Performance of systems**]: *Modeling techniques*

General Terms

Performance, Theory

Keywords

Virtualization, consolidation, energy modeling, emulation.

1. INTRODUCTION

In the last decade many large-scale services have migrated to cloud infrastructures, creating an equal increase in virtualized data centers. Data center infrastructures have become one of the largest and fastest growing consumers of electricity globally, surpassing the aviation industry both in terms of energy consumption and CO_2 emissions [5]. To put this into perspective, in 2013, U.S. data center's electricity consumption (91 TWh) was sufficient to power twice the number of all the households in New York City [22]. As a result, ICT energy consumption accounts for 3% of the global consumption and has an annual increase of ≈4.3% [29]. Consequently, there is a growing interest to improve energy efficiency in data center's design, with obvious environmental and financial motives.

A first approach towards building greener ICT was the development of *energy proportional* computing and networking infrastructures [6, 17]. This effort took advantage of energy efficient hardware, like CPU voltage/frequency scaling and sleep states, low-power Ethernet and power-efficient OS-level resource management (e.g. on demand Linux governor and PowerNap [7]). However, even at low utilization loads, in the order of 10%, the server power consumption can reach up to 50% of its peak demand [10], allowing room for further improvement. To further reduce energy consumption, research has developed workload consolidation algorithms which concentrate computation into a subset of the data center infrastructures.

Host and OS virtualization have enabled additional power consolidation techniques, providing support for virtual machine (VM) migration. VM migration allows seamless relocation of VMs between physical hosts, with relatively short down-times. VM migration allows service providers to dynamically aggregate load on fewer physical hosts, while fulfilling a minimum guaranteed level of performance, expressed in the form of service-level-agreements (SLA). VM consolidation strategies commonly transform the placement algorithm into an optimization problem, using as constraints the estimated VM resource requirements and the available resources of the physical hosts. The evaluation of the proposed approximation algorithms is typically based on custom simulation frameworks [3, 12] or small-scale testbeds [9, 24, 28].

In this paper we challenge common practices used to design and evaluate VM consolidation strategies. We argue that a set of important system parameters is commonly ignored in favor of simplicity, namely:

- the dynamic **energy consumption** profiles of servers;
- the complexity in **resource sharing** between VMs in a single host (*e.g.,* CPU, disk, network, memory), as well as the overhead of virtualization;
- the performance characteristics of the underlying **network** infrastructure (topology, speed, configuration) and the employed network protocols;
- the **cost** of live **VM migration** in terms of energy, network traffic and application-level performance;
- complex performance behaviors of networked systems observed in **large scale** deployments.

This paper argues that underestimating the impact of the aforementioned system properties in the evaluation of VM consolidation algorithms introduces significant inaccuracies. Hence, existing algorithms make relocation decisions based on inaccurate performance predictions for co-hosted VMs, as well as they overlook migration overheads. In addition, common evaluation methods not only ignore these properties, but also rely on very small scale experiments.

In an effort to address the aforementioned issues, we point out how existing solutions can be reused, combined and extended in order to create an evaluation framework that allows the reliable exploration of the energy-performance trade-offs in VM consolidation strategies. Such a solution is particularly useful, since only few researchers can access a real-sized data center infrastructure for experimentation.

We initially present related work in the field (§ 2) and then highlight that many potential pitfalls exist in the methods used to model application performance and energy requirements of data center servers (§ 3). Furthermore, we propose a new methodology that is funded on two components: (i) the measurement-based characterization of physical servers, and (ii) the emulation-based estimation of the load offered by VMs and management tools to the host servers under different configurations (§ 4). Finally, we conclude by summarizing this work (§ 5).

2. BACKGROUND AND RELATED WORK

VMs placement in a single data center infrastructure is a compound problem with multiple competing objectives. Firstly, *consolidation* aims to compress workloads into as few physical hosts as possible, and either turn off or leave idle the unused part of the infrastructure. During this step, the objective is to maximize the energy saving, at the cost of performance. Second, the opposite to the process of consolidation, is the *elimination of performance hot-spots* (*e.g.,* Sandpiper [28]), which spreads VMs across the data center, increasing the active physical hosts. Lastly, a *load-balancing* process can run in the background and relocate VMs aiming to smoothen the load variations across the infrastructure, and therefore, better absorb the performance spikes of bursty workloads.

Usually hot-spot removal and consolidation are used together, hand by hand. The two functionalities have opposing goals, but are equally necessary to achieve an equilibrium between performance and energy saving. Specifically, this is the most important aspect in designing greener data center solutions: *making informed decisions about the application-level performance which is sacrificed in trade for lower energy consumption, and vice versa.*

Energy-efficient VM placement algorithms: The energy/performance trade-off is controlled by the VM placement algorithms, which implement the decision-making logic for the followings:

- **Choose a source host** with average utilization above, in case of hotspot removal, or below, in a case of consolidation, a threshold.
- **Choose a VM** from the selected host based on the resources it requires. For example, during the evacuation of an under-utilized server, VMs are ordered based on their resource requirements.
- **Choose a destination host** with sufficient available resources (*e.g.,* disk, network, CPU, memory) meeting the minimum performance guarantees for the application which runs in the guest (determined from SLAs).

A significant number of research efforts reduce this decision making process into a vectorized bin packing problem [3, 20]. VMs are represented as n-dimensional vectors of estimated resource demands, while each host is represented as an n-dimensional vector of available resources. VM placement aims to fulfill the minimum guaranteed resources, as specified through service SLAs, while minimizing the number of active hosts. Since the vector bin packing problem is NP-hard, a number of near-optimal solutions have been proposed using a variety of heuristics [26, 27] (*e.g.,* first-fit decreasing, best-fit decreasing, worst-fit decreasing, etc.). Alternative approaches towards the placement problem use genetic algorithms [16] and dynamic programming [12].

Unfortunately, the common methodology for evaluating the above solutions is simulation, which abstracts important properties of virtualized systems, discussed in detail in Section 3. As a result their applicability on real data center environments is limited. This observation is also supported by studies like [26, 28], that approach the problem from a more practical perspective. Other works like [9, 24], evaluate their systems using small testbeds with no more that few tens of machines, insufficient to capture the scalability of the resulting system in real-sized environments.

3. COMMON PITFALLS OF VM PLACEMENT ALGORITHMS

The VM migration decisions use as inputs: (i) the resource requirements of a VM (given an SLA), (ii) the expected load increase in the destination host, (iii) the available resources of the physical hosts, and (iv) the expected level of performance for VM applications. The main argument of this paper states that the majority of the existing works does not accurately capture the aforementioned decision criteria. Relevant research efforts typically employ simplistic models to decide the placement of VMs and they are evaluated using model simulation or small-scale testbeds. As a result, they fail to capture important aspects of application-level performance and energy consumption.

In this section we elaborate on the aforementioned evaluation and design pitfalls, related to the specific properties of large-scale virtualized data centers which tend to ignore: (i) the dynamic of the underlying resource sharing model and the migration cost (§ 3.1) and (ii) the energy consumption profiles under mixed workloads (§ 3.2).

3.1 Shared virtualized resources and application-level performance

Cloud providers have been refrained from using consolidation algorithms on their infrastructures mainly because it is not easy to predict the performance penalties on hosted applications. While the overhead of virtualization has been significantly reduced (*e.g., paravirtualized* I/O, hardware support), the interaction model with a host's physical resources has become more complex. For example, Wang *el al.* [11], exemplify some interesting artifacts in the perceived CPU and network resource availability by guest OS. Such performance variability has been measured to significantly affect large-scale time-sensitive services [15].

This performance variability is a direct consequence of the resource sharing functionality implementation between co-hosted VMs. Nevertheless, most of the heuristics used in VM placement algorithms, assume that the virtualization platforms provide perfect performance isolation. Hence, they suggest that VM resource utilization, and consequently application-level performance, remains the same across different hosts.

The above assumption, however, is very simplistic and can lead to incorrect VM placement decisions. The amount of the resources which each VM receives depends on three factors: the scheduling policy of the hypervisor, the available resources of the hosting platform, and the activity of co-hosted VMs. None of these three factors can be considered static, and moreover, they exhibit a high degree of inter-dependencies. For example, consider many highly-utilized VMs collocated on a server, each receiving a fair share of the CPU time. On a lower utilized server, the same VM would almost certainly reach a higher peak. Therefore, a typical hot spot removal algorithm would underestimate the peak CPU requirements of a VM, and could potentially make the sub-optimal decisions.

Another over-simplifying assumption which is commonly made, is the inference of application SLA violations, based on VM or host-level utilization metrics. First, the poor resource sharing models which are used during evaluation, do not provide accurate utilization estimations. Second, it is fairly unreliable to employ only the CPU utilization to infer SLA violations, since this approach is susceptible to false negatives, especially for bursty workloads. This problem has been pointed out by *Wood et al.* in [28], via extensive experiments.

Finally, the available network resources is another important factor which also determines the application-level performance. This includes the available bandwidth at the end-hosts (including the CPU overheads of packet processing), the employed protocols, the topology of the data center's networking infrastructure, the speed of physical links, and the scheduling algorithms at intermediate devices. None of these important characteristics are sufficiently replicated in the majority of the VM consolidation studies. The simulation frameworks use an over-simplified model for the network, and the small testbeds fail to reproduce the complex behaviors observed in large scale networked systems.

3.2 Accuracy of energy consumption models

Many of the VM placement approaches, covered in § 2, provide only gross insights on the achieved energy saving, when we aggregate the workload on fewer hosts. The achieved accuracy in the estimated savings is usually restricted at the level of accounting the number of powered on servers over the unit of time. This reduced level of detail does not allow users to effectively evaluate the energy-performance trade-off.

Some research efforts, like [2, 3], consider the use of a more detailed energy model. Effectively, they are based on the observation that CPU utilization is highly correlated with the overall energy consumption of a server. As a result, they use linear models which are based on current utilization levels to estimate the energy consumption. Normally, these models have to be fitted separately for each type of server. While this effort is certainly a step towards the right direction, it has known limitations. First, it ignores the impact of the CPU DVFS features and does not account the energy consumption of other hardware components like disk, memory and network.

The importance of the above facts has been pointed out by several studies (*e.g.,* [1, 14]), showing that depending on the characteristics of a workload, the level of CPU-load alone might not be a very accurate metric. This is especially true for storage and network devices which implement energy saving features, and therefore have also a wide dynamic energy range. Second, following the discussion of the previous section, system utilization is not modeled accurately in the simulation frameworks which are commonly used to evaluate VM consolidation algorithms. Hence, the input which is used in their linear energy/CPU-utilization models, is not reliable.

Ideally, we would like to have evaluation frameworks which can replicate with good accuracy the load of different system components over time, generated from custom scenarios of migrating VMs who execute a given workload. Thereafter, accurate heuristic models could be applied to estimate the energy consumption of servers using as input the collected measurements for the CPU-load as well as the disk, the network and the memory I/O operations.

3.3 Live VM migration does not come for free

Live VM migration is a first-class citizen in the data center energy consolidation problem because of the unprecedented level of flexibility it offers. While useful, it is a complex process with two main approaches. In the *pre-copy* [4] method, the hypervisor first copies all memory pages of the VM to the destination host, and while they change (become "dirty") they are re-copied until the dirtying rate slows down. The second method, post-copy [19], performs the inverse functionality; VM is first suspended temporarily, a minimal state (*e.g.,* CPU state, registers) is transferred and then resumed at the target host. At the same time memory pages are pushed to the target (pre-paging), and whenever a page fault occurs the missing page is fetched from the source on demand.

From the above it is clear that the live migration of a VM, introduces overheads at multiple levels. First, it increases the CPU load on the management domain of both the source and the target host, second it creates extra network traffic, and finally it temporarily degrades the performance of the applications which run on the migrating VM. The intensity of these overheads heavily depend on the characteristics of the applications, the offered workload, and on the available CPU and network resources. These facts have been pointed out from multiple empirical studies, like [13] and [23].

Unfortunately, very few of the existing VM consolidation studies take into consideration the aforementioned overheads. Even those who do (*e.g.,* [2, 3]), they use analytical models which not only are simplistic, but they also rely on input which is not accurately replicable by the employed validation frameworks. As a result, such models commonly overlook the cost of migration (especially at scale) resulting in highly over-optimistic and unrealistic results. Also, the inability of these approaches to capture the effects of VM migration on resources utilization, further incommodes the study of the energy-performance trade-off for different VM placement algorithms.

4. PROPOSED METHODOLOGY

The objective of this section is to address the main points of criticism on past works by proposing practical solutions which provide an environment to faithfully evaluate energy-efficient VM placement algorithms. An accurate and effective experimentation framework should incorporate a unified model for the available resources of a virtualized server, the demands of hosted applications, the properties of network infrastructure and the energy consumption of devices based on the load. Ultimately, such a solution will allow users to benchmark their ideas and efficiently explore the important trade-off between energy saving and application-level performance penalties.

Motivated by recent efforts in the faithful replication of experiments for networked systems (§ 4.1), we elaborate on ways to integrate energy consumption models (§ 4.2).

4.1 Data-center modeling platform

In the recent years, a resurging interest has surfaced aiming for reproducible network experimentation. This has given rise in the development of generic network experimentation platforms, which allow seamless replication of large scale systems. Mininet [21] was a pioneering tool, offering an emulation platform with support for dense topologies, using Linux containers and network namespaces. Overcoming the limitations of simulation, users could now reuse real applications and OS components to recreate topologies and experiment scenarios via a script-based automation interface. A more elaborate effort aiming to overcome Mininet's poor scalability, is SELENA [8]. SELENA's design employs *time dilation* as a way to improve the experimental fidelity at scale while maintaining reproducibility across different platforms.

Hereafter, we describe how SELENA can be used and extended, forming the basis for a data center modeling platform which fulfills the requirement set in the previous section. It should be emphasized that the proposed approach is not tightly coupled with SELENA and it is compatible with other emulation frameworks which rely on virtualization and provide resource management primitives.

Reusing applications, emulating VMs: SELENA is a Xen-based[1] emulation framework, thus supports the re-use of unmodified code and common OSes. VMs can be configured to form virtual networks (in-a-box) and recreate the properties of real networks (*e.g.,* topology, link speed, latency) and real hosts (*e.g.,* OS configuration, network protocols, resources, etc.). Nonetheless, SELENA employs a one-VM-per-host mapping, which can limit effective scaling of data center-scale experiments. This can be addressed by abstracting a subset of less important data center nodes, using more lightweight hybrid approach, such as using Mininet inside a time-dilated SELENA VM. A second extension is to create an abstraction to represent the entity of a VM instance which runs on a server. Instead of using a heavyweight approach such as nested virtualization, this can be implemented by using containers inside SELENA VMs. Finally, a new mechanism for grouping VMs on hosts, hosts on racks and racks in pods, is also a useful feature, particularly during the design of VM placement controllers.

Fidelity at scale: In order to faithfully emulate faster and larger computer networks, SELENA's technique of time-dilation transparently slows down the passage of time for guest operating systems. It effectively virtualizes the availability of hosting hardware resources and therefore, allows the recreation of scenarios with increased I/O and computational demands. Users can directly control the trade-off between fidelity and running-time via intuitive tuning knobs. To further improve SELENA's scalability, we could explore zero-copy inter-guest network connectivity [18] and also the distributed execution of experiments across multiple hosts.

Emulating resource utilization: SELENA relies on the Xen hypervisor which provides by design the primitives to finely share host resources between VMs. We identify four key virtual resources: CPU cycles, memory, disk and network I/O operations. The hypervisor allocates CPU resources between VMs using the *credit2* scheduler, a highly flexible and tunable scheduler. Memory resources are abstracted by the hypervisor using a grant table access control mechanism, which enables accurate memory allocation to each VM. The support of the Xen hypervisor for disk I/O rate control is limited to simple inter-VM prioritization. Nevertheless, Linux *cgroups* (via the *blkio* controller), allows users to regulate the rate of I/O operations allowed in a unit of time. In a Xen environment, a user can use *cgroups* based throttling from inside the guests. Network I/O rates can be controlled either from the VIF QoS primitives offered by the Xen *netback* driver, or from within the guest by using *tc* on a virtual interface's egress queue.

Using the above mechanisms, we can determine with a greater level of control the maximum amount of resources which are available to a VM, or to groups of VMs. The latter is particularly useful because many VMs will be grouped under the common abstraction of a host, hence, they need to share a common pool of resources. The maximum amount of available resources to each group, will be equal to the available resources of the real system's components we want to model. Time dilation, on the other hand, will help to virtually scale the resources of the emulation machine, and support larger experiments. With the described extensions and using different workloads, the utilization levels of emulated resources, and the performance of applications (running inside the regulated VMs), will approximate reality with substantially higher accuracy in comparison to a simulation.

Emulating the cost of live VM migration: Multiple studies have tried to analyze and model the impact of live VM migration in terms of application-perceived performance degradation, network resource requirements [23], as well as energy cost [13], migration duration and down time of a migrating VM [25]. Since our experiments execute on top of a single Xen hypervisor instance, it is not possible neither scalable to perform actual migrations. Therefore, our intention is to emulate the process of a pre-copy live VM migration inside SELENA in a lightweight way.

In order to emulate a migration, we can employ any of the aforementioned models and given the dirty page rate of a migrating VM, we can artificially recreate during runtime the followings: (i) the migration-specific network traffic volume, (ii) the VM downtime, and (iii) the extra CPU load. Since the resources which are available to each VM are regulated to match the real system (see above), the proposed methodology will accurately replicate the extra load introduced from migration. Consequently, the impact of a migration on the running application's performance will also be captured.

4.2 A system-load based energy model

Earlier measurement-based evaluations [1, 14] show that there is a huge impact of the energy management/enhancement techniques on system power requirements. So far, this impact has not been addressed by VM migration or consolidation strategies. As an example, by reproducing the methodology proposed in [14] in our testbed, we show in Figure 1 the power consumption of the CPU of one of our data center servers, `quorum-102`, versus the load expressed in active cycles per second, namely ACPS. In the figure we observe

[1] http://www.xenproject.org/

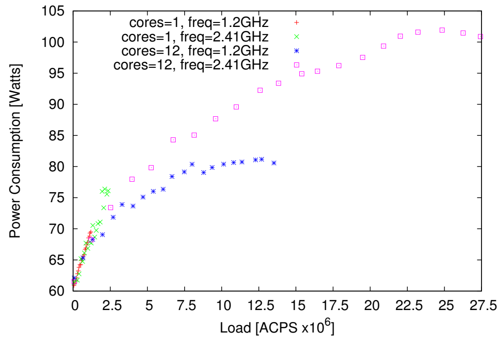

Figure 1: CPU performance bounds of `quorum-102`.

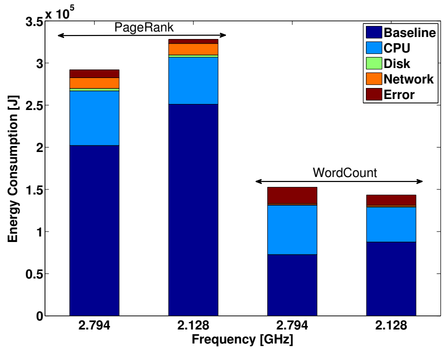

Figure 2: Energy consumption of our servers in a cloud-based scenario.

that changing the number of cores and the CPU frequency produces rather different power consumption levels. More specifically, the authors of [14] have shown that energy consumption and efficiency of each server component can be accurately estimated upon a statistical characterization of baseline server energy consumption plus CPU utilization, disk I/O activity, and the network activity. Since CPU loads due to different server components operation are additive, the resulting total energy consumption of the server is the sum of the individual components' consumption, as experimentally shown in [14].

Therefore, collecting activity patterns of VMs is key to estimate the energy behavior of a modeled real system under VM consolidation and live migration strategy operation. As we discussed in § 3.2, to attain a reliable estimate of energy requirements for data center servers we need to obtain usage information for individual components. Indeed, we can access such information through the Xen hypervisor, which maintains fine grain accounting of the usage statistics for all computational resources: *i.e.,* CPU, disk I/O, memory I/O, and network I/O. However, it is important and challenging, to calibrate those statistics so that they refer to the load of a real system and not to the *virtual* load of the VM. This, though, is guaranteed through the resource utilization emulation model, discussed in § 4.1. Furthermore, such statistics can be emulated and used as input to a utilization-based energy estimation model. Like in [14], such model can be built using measurements from real server-grade machines, whereas emulation can be suitably used to estimate the load.

Energy model details. For the implementation of the utilization-based energy model, we propose to use the methodology described in [14]. We need however to extend it to include estimates for memory energy utilization (which was included in the *baseline* component in that work), and build an energy-performance model. Specifically, by collecting power and activity measurements for CPU, disk, memory and network of real servers, we can generate accurate functions which approximate the energy requirements.

Our per-component energy model depends on a few activity parameters, which are the active CPU cycles per second, number of read/write disk and memory operations, and network utilization. Moreover, the effect of multicore processors and DVFS is not to be neglected, as can be seen in Figure 1, since both characteristics yields high variability in the energy consumption. Finally, we also consider a residual *baseline* energy consumption, which represents the activity of the server when no user-level process is active.

In what follows, since it is possible to characterize the CPU activity due to each different server component, we remove the energy consumption due to CPU activity in the estimate of each component's energy consumption. Thereby, energy requirements can be expressed as follows:

$$\begin{aligned} E_{cpu} &= \eta_{cpu}(T, f, c, a), && (1)\\ E_{disk} &= \eta_{dr}(T, f, c, cs, nc) + \eta_{dw}(T, f, c, cs, nc), && (2)\\ E_{net} &= \eta_{in}(T, f, c, s, l) + \eta_{out}(T, f, c, s, l), && (3)\\ E_{mem} &= \eta_{mr}(T, f, c, cs, nc) + \eta_{mw}(T, f, c, cs, nc); && (4) \end{aligned}$$

where $\eta_j, \forall j \in \{cpu, dr, dw, in, out, mr, mw\}$ is the efficiency of the CPU, the disk while reading or writing, the network while receiving or sending and the memory while reading or writing, respectively. T defines the duration of the experiment, f is the system frequency, c is the amount of cores used by the system, a is the CPU activity expressed in active cycles per second, cs is the chunk size used to read/write from/to the disk or memory, nc is the total number of chunks which has to be read/written from/to the disk or memory, s corresponds to the packet size flowing over the network and l is the network load. In the above model, we consider that network transmission and reception are independent processes. The same applies for the electro-mechanical operation of disks for *read* and *write*. Similarly, the behavior of memory is independent for *read* and *write* events. Since we do not account for CPU active cycles in all those events (the cost of CPU cycles is computed separately), we can safely assume that *read* and *write* operations of memory, disk and network are "energy-orthogonal", *i.e.,* they do not share energy consumption, and therefore we can simply sum up the respective energy consumptions for each component.

The resulting total energy estimation of the system is:

$$E_{total} = E_{base} + E_{cpu} + E_{disk} + E_{net} + E_{mem} \quad (5)$$

where $E_i, \forall i \in \{total, base, cpu, disk, net, mem\}$, corresponds to the energy requirements for the whole system, the baseline, the CPU, disk, network and memory respectively.

However, to be able to use the model in the evaluation of VMs consolidation and/or migration strategies, we also need to characterize the load of VMs on different machines, possibly using different hardware and configurations. Therefore, we need to emulate the activity of VMs and VM management software to estimate the correct amount of load caused to the host machines (*e.g.,* before and after migration). With SELENA, all we need to do is to use the set of counters for usage statistics regarding the activity of the VMs and "feed" those statistics to the model described above. In addition, statistics of VMs which are grouped under the same "server" virtual entity (implemented as a common pool of resources), should be aggregated.

Preliminary evaluation. To evaluate the validity of the presented energy model, we set up a small scale experiment. We measure two similar servers, `quorum-101` and `quorum-102`, for consistency reasons. The servers are Dell PowerEdge R320 (12th generation) with Intel Xeon E5-2430L V2 2.4GHz (6 cores), two hard drives, a 100 *GB* SSD and a 1 *TB* HDD, two *Gigabit* and two 10 *Gigabit* ports. We have installed the Linux Ubuntu Server 14.04LTS and also a recent version of the Xen hypervisor (v4.4).To monitor the instantaneous power consumption of the system we use the Sentry CDUs[2]. We collect our measurements every second via the `snmp` protocol and we store them locally.

In Figure 2 we show some preliminary results creating workloads on virtual machines. In the figure we can see the estimation of the per-component energy consumption, expressed in *Joules*, for one of our servers considering the effect of DVFS. For the evaluation we use a cloud scenario with two servers, each one hosting a VM. In this scenario we run two Hadoop applications, WordCount and Pagerank algorithm, and we keep track of the instantaneous power consumption and the overall utilization of CPU, disk and network for two different frequencies of the server. As can be seen in the figure, the two applications have different utilization profiles for individual system components. It is worth mentioning that we have simplified our model, including the effect of memory within the other components (but it will be considered separately in later stages of our study).

Importing the utilization results into the model we estimate the accumulated energy consumption for the server which runs the application. From the power measurements we can extract the actual energy needs and finally, we observe that the estimation error is on average about 4% and never worse than 10%. We expect that this error can be reduced when we properly include the memory behavior.

5. CONCLUSION

This paper challenged common evaluation practices employed in past VM consolidation studies, such as simulation and small testbeds, which fail to capture the fundamental properties of real systems. Specifically, we identified a series of over-simplifying assumptions regarding energy consumption and performance characteristics with respect to virtualized infrastructures. To address this problem, we described the design of an evaluation framework which incorporates more accurate models for data center systems and their available resources. In addition, we proposed a measurement-based power characterization methodology for servers, which accepts as input the load of individual hardware components and estimates the energy consumption for different server configurations. The integration of the two solutions, allows us to achieve the envisioned goal of exploring the energy-performance trade-off in VM consolidation.

6. REFERENCES

[1] A. Gandhi *el al.*. Optimal power allocation in server farms. In *SIGMETRICS Performance Evaluation Review*, volume 37. ACM, 2009.

[2] A. Verma *el al.*. pMapper: power and migration cost aware application placement in virtualized systems. In *Middleware*. Springer, 2008.

[3] A. Beloglazov and R. Buyya. Energy efficient resource management in virtualized cloud data centers. In *MGC*. IEEE, 2010.

[2]Sentry Sw.Cabinet Distribution Unit CDW-24VEA458/C

[4] C. Clark *el al.*. Live migration of virtual machines. In *NSDI*. USENIX, 2005.

[5] C. Pettey. Gartner estimates ICT industry accounts for 2% of global CO_2 emissions. http://goo.gl/4KuOAi, 2007.

[6] D. Abts *el al.*. Energy proportional datacenter networks. In *SIGARCH Computer Architecture News*, volume 38. ACM, 2010.

[7] D. Meisner *el al.*. Powernap: eliminating server idle power. *SIGARCH Computer Architecture News*, 37(1), 2009.

[8] D. Pediaditakis *el al.*. Faithful reproduction of network experiments. In *ANCS*. ACM, 2014.

[9] F. Hermenier *el al.*. Entropy: a consolidation manager for clusters. In *VEE*. ACM, 2009.

[10] G. Chen *el al.*. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. In *NSDI*, volume 8, 2008.

[11] G. Wang and T. S. E. Ng. The impact of virtualization on network performance of amazon ec2 data center. In *INFOCOM*. IEEE, 2010.

[12] H. Goudarzi *el al.*. SLA-based optimization of power and migration cost in cloud computing. In *CCGrid*. IEEE, 2012.

[13] H. Liu *el al.*. Performance and energy modeling for live migration of virtual machines. *Cluster computing*, 16(2), 2013.

[14] J. Arjona Aroca *el al.*. A measurement-based analysis of the energy consumption of data center servers. In *e-Energy*. ACM, 2014.

[15] J. Dean and L. A. Barroso. The tail at scale. *Commun. ACM*, 56(2), Feb. 2013.

[16] J. Xu and J. AB Fortes. Multi-objective virtual machine placement in virtualized data center environments. In *GreenCom*. IEEE, 2010.

[17] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *IEEE computer*, 40, 2007.

[18] L. Rizzo and G. Lettieri. VALE, a Switched Ethernet for Virtual Machines. In *CoNEXT*. ACM, 2012.

[19] M. R. Hines *el al.*. Post-copy live migration of virtual machines. *ACM SIGOPS operating systems review*, 43(3), 2009.

[20] N. Bobroff *el al.*. Dynamic placement of virtual machines for managing SLA violations. In *IM*. IEEE, 2007.

[21] N. Handigol *el al.*. Reproducible network experiments using container-based emulation. In *CoNEXT*. ACM, 2012.

[22] P. Delforge. America's data centers consuming and wasting growing amounts of energy. http://goo.gl/HOLLBx, 2014.

[23] R. Bradford *el al.*. Live wide-area migration of virtual machines including local persistent state. In *VEE*. ACM, 2007.

[24] R. Nathuji *el al.*. VPM tokens: virtual machine-aware power budgeting in datacenters. *Cluster computing*, 12(2), 2009.

[25] S. Akoush *el al.*. Predicting the performance of virtual machine migration. In *MASCOTS*. IEEE, 2010.

[26] S. Lee *el al.*. Validating heuristics for virtual machines consolidation. *Microsoft Research, MSR-TR-2011-9*, 2011.

[27] T. C. Ferreto *el al.*. Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, 27(8), 2011.

[28] T. Wood *el al.*. Black-box and gray-box strategies for virtual machine migration. In *NSDI*, volume 7, 2007.

[29] W. Van Heddeghem *el al.*. Trends in worldwide ICT electricity consumption from 2007 to 2012. *Computer Communications*, 2014.